



ENOLA

SysML for Reviewers

October 2024

About US

LEAVE THE PAST IN THE PAST

IT'S TIME TO EVOLVE, DIGITALLY

WE ARE A TRAINING, COACHING, AND CONSULTING FIRM DEDICATED TO THE RAPID EVOLUTION OF OUR CLIENTS WITHIN THE DIGITAL UNIVERSE. OUR SERVICES ARE DIRECTED TOWARDS:

- DIGITAL ENGINEERING/TRANSFORMATION
- MODEL BASED SYSTEMS ENGINEERING
- ENTERPRISE ARCHITECTURE
- SOFTWARE ARCHITECTURE
- DATABASE ARCHITECTURE
- ONTOLOGIES
- COLLABORATION SERVER MANAGEMENT

MISSION

Enola WILL train, coach, and mentor your staff to be independently successful as quickly as possible.

Yes, our mission is to work ourselves out of a job!

COURSE DESCRIPTION

The SysML for Reviewers Course is a two-day training designed to provide personnel with an understanding of the Systems Modeling Language (SysML) and Cameo Collaborator to proficiently contribute to model-based reviews.

This course provides a mix of slides and instructor-led demonstrations. Our trainers are all experienced practitioners who understand the balance of theory and practicality.

Prerequisites:

None

Required Software:

TBD

Take-Aways:

- Understanding SysML for the perspective of a Reviewer
- Working knowledge of Cameo Collaborator

AGENDA

DAY 1:

- Introduction to SysML
- Requirements, Structure, Behavior, Parametrics
- Introduction to Cameo Collaborator
- Model Structure
- Package Diagrams (PKG)
- Requirements
- Requirements Diagrams and Tables (REQ)
- Structure
- Block Definition Diagrams (BDD)
- Internal Block Diagrams (IBD)

DAY 2:

- Behavior
- Use Case Diagrams (UC)
- Activity Diagrams (ACT)
- Sequence Diagrams (SD)
- State Machines (STM)
- Parametrics
- Constraints and Parametrics (PAR)

COURSE CONTENT

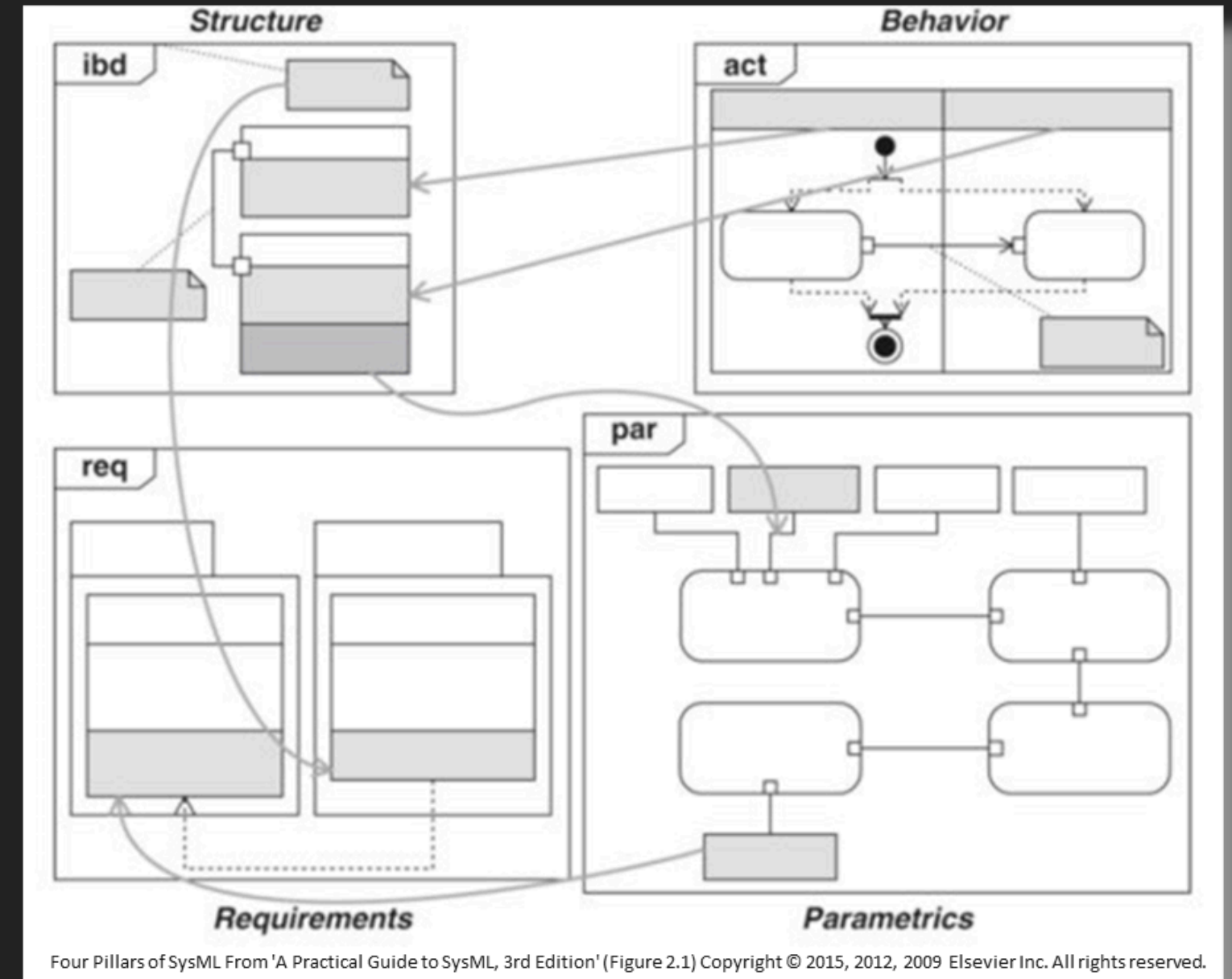


Introduction to SysML

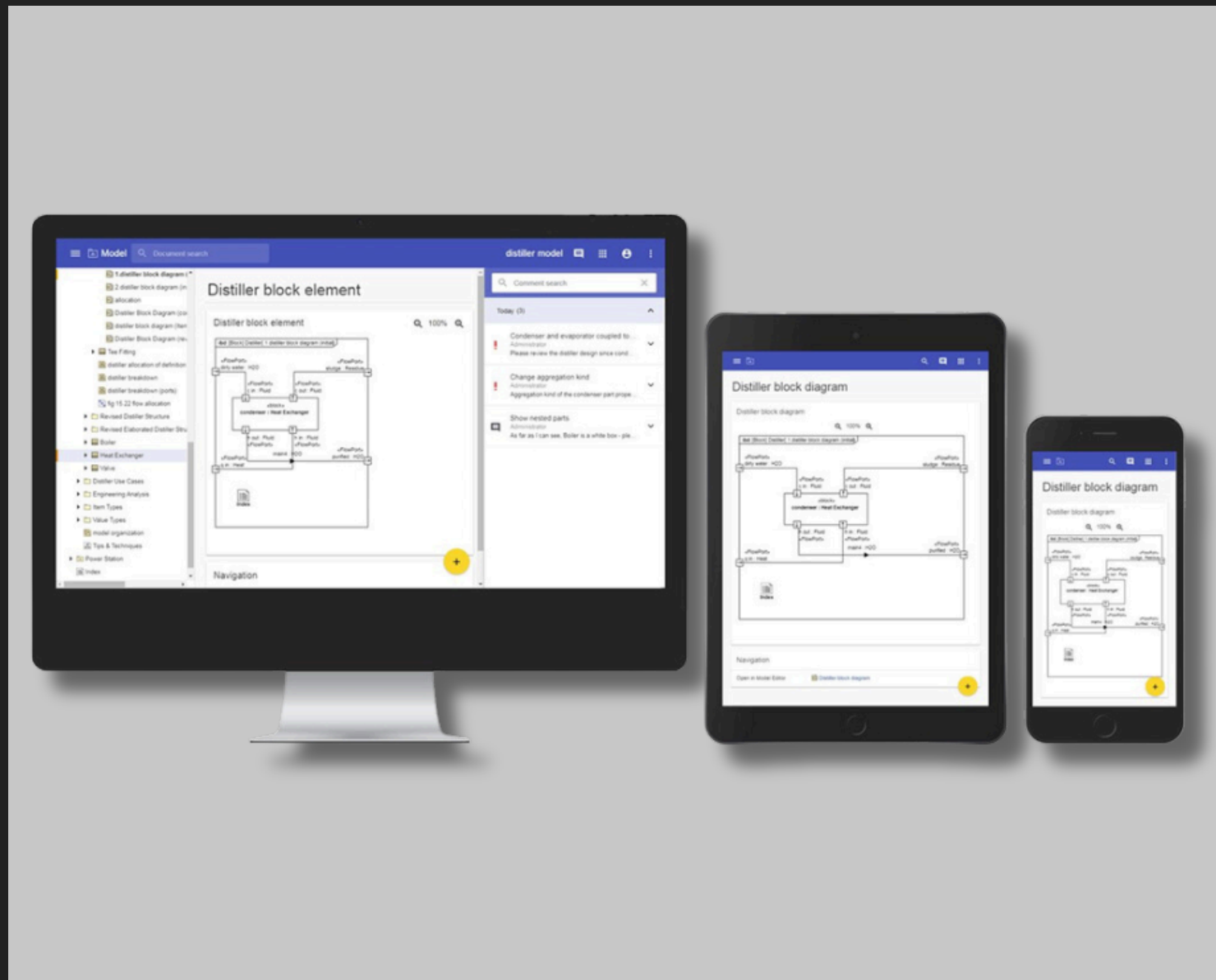
•MBSE requires users to choose a language, tool, and methodology. This course focuses on the language (SysML) and touches on the tool for reviewing models (Cameo Collaborator).

This module covers:

- Key Definitions
- Introduction to MBSE
- Introduction to SysML



Introduction to COLLABORATOR



Cameo Collaborator is a web-based product that allows stakeholders and teams to leave model feedback via textual or graphical comments, perform limited model editing, view on web or in the model, and utilize the same role-based access as Teamwork Cloud.

This module covers:

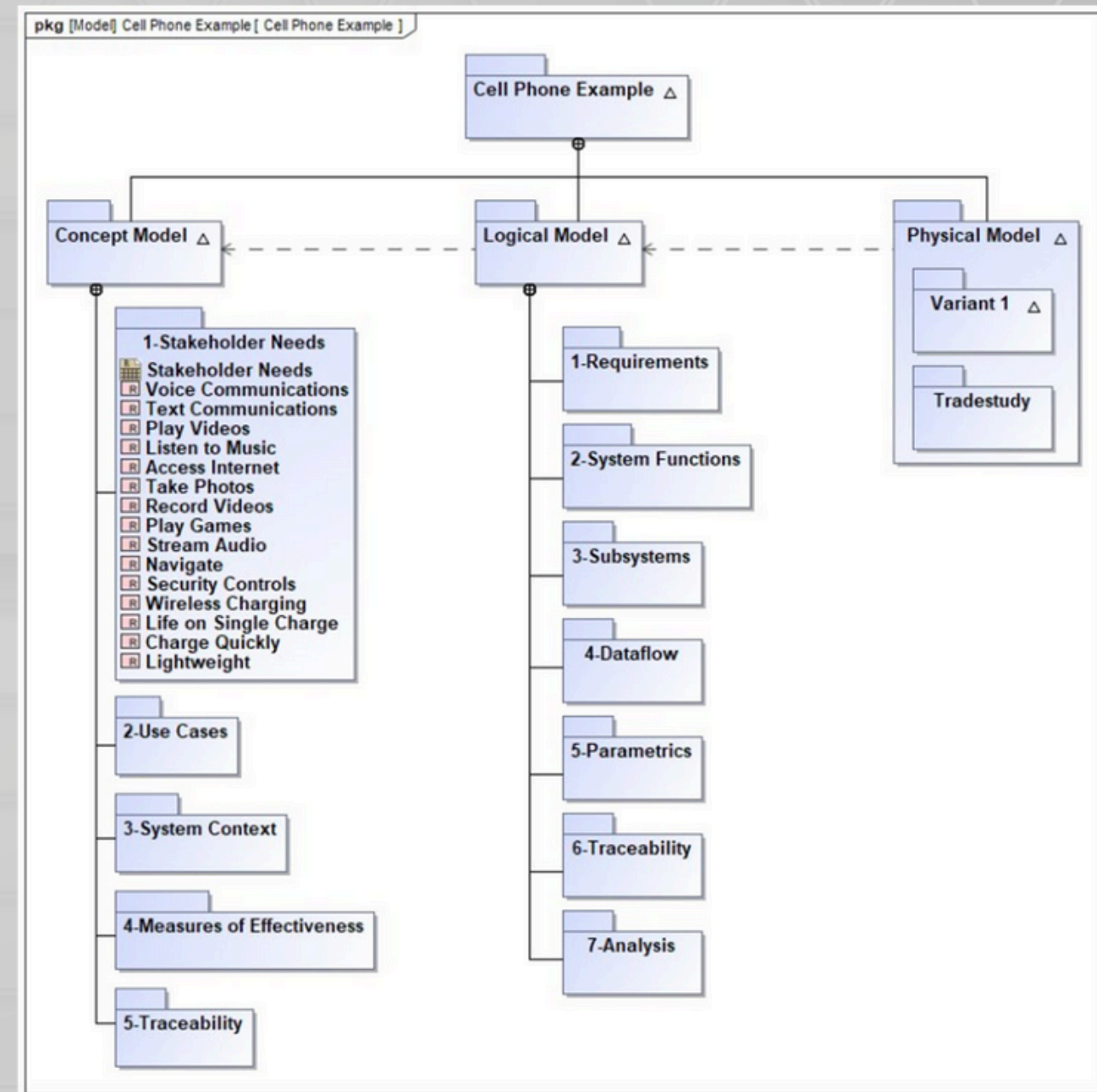
- What is Teamwork Cloud?
- Teamwork Cloud Overview
- Cameo Collaborator Overview
- High-Level Architecture
- Collaborator User Interface
- Reviewing Models with Collaborator
- Collaborator Comments
- Model Editing with Collaborator

Packages

Packages, Models, Libraries and Profiles enable users to define the structure of the database to improve navigability and data grouping.

This module covers:

- Diagram Example and Purpose
- Packages
- Containment
- Models
- Model Libraries
- Diagram Annotations
- Project Structure Best Practices

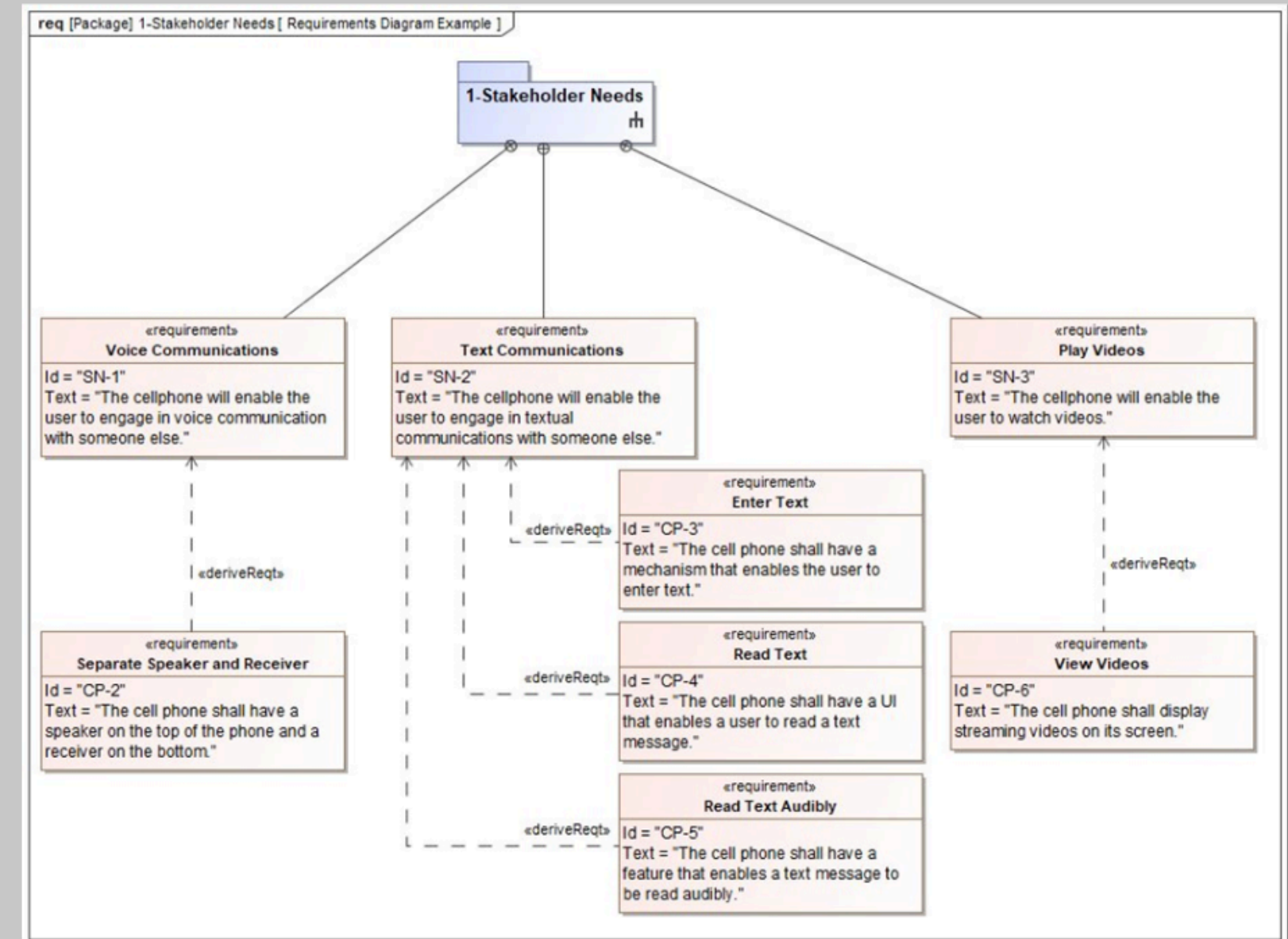


Requirements

Requirements are text-based statements that must be met for a system to deliver the required functionality within certain performance metrics.

This Module Covers:

- Diagram Example and Purpose
- Requirements and Extended Requirements
- Requirement Abstractions
- Requirement Relationships
- Requirement Tables, Matrices, and Maps

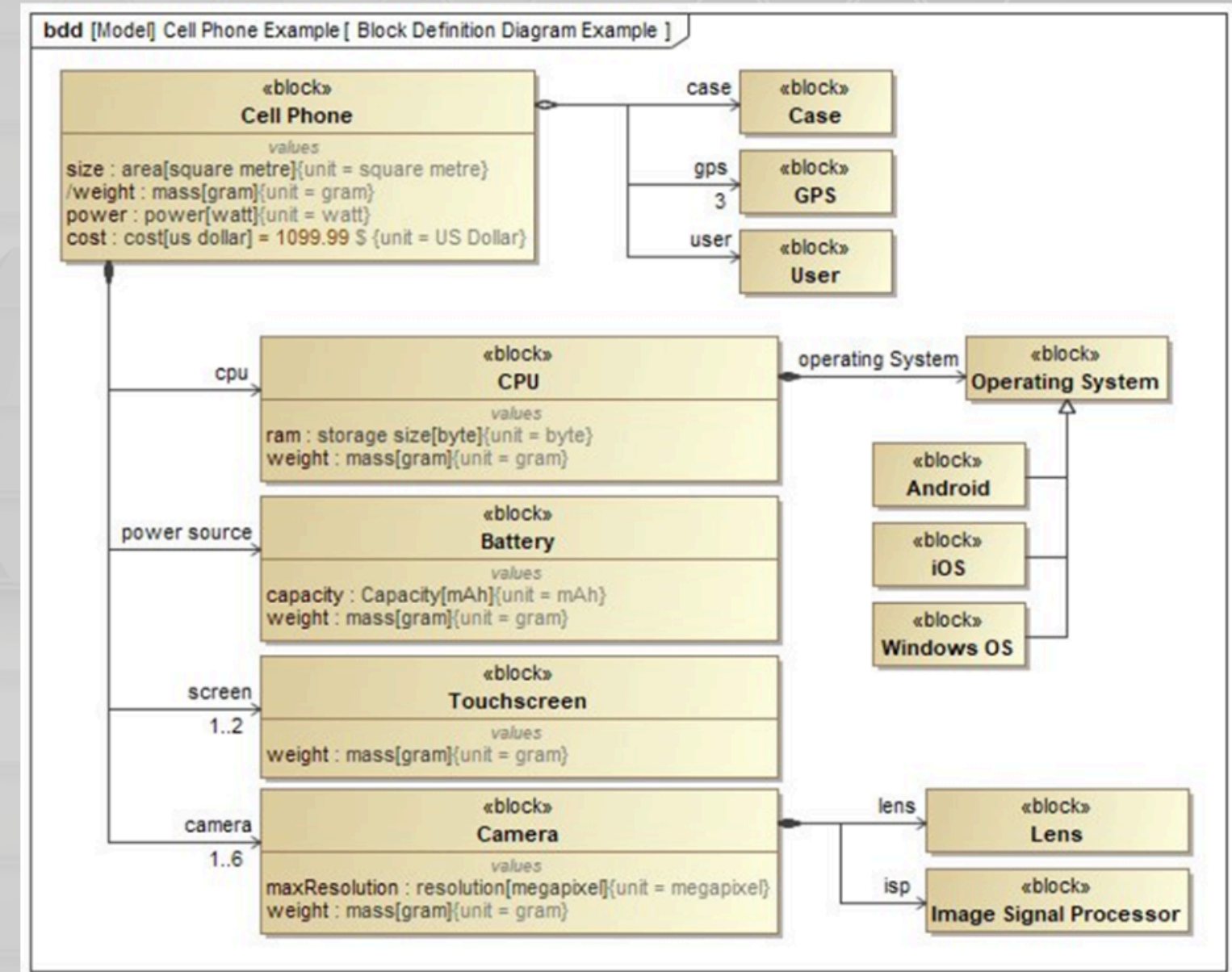


Block Definition Diagrams

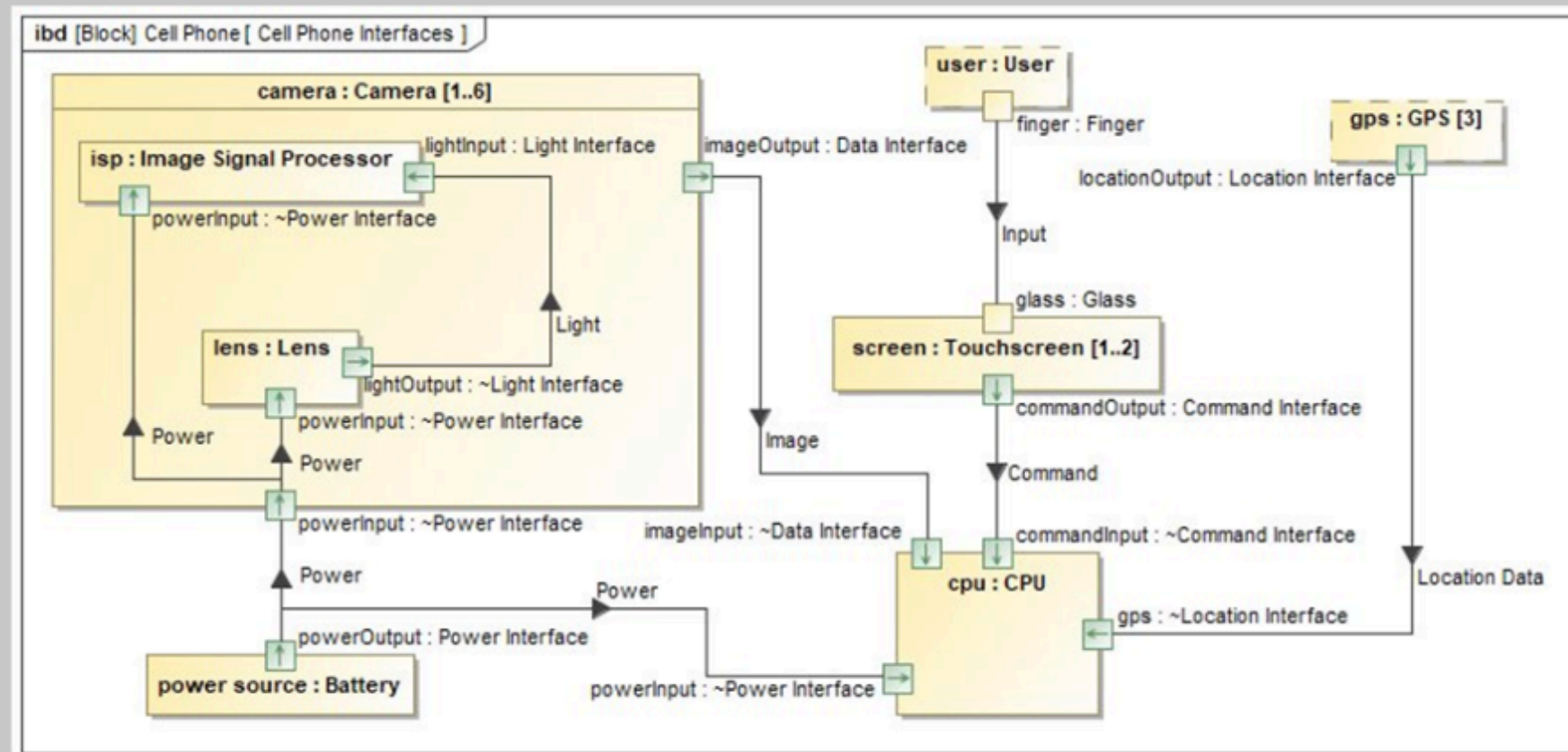
While Block Definition Diagrams (BDDs) have many uses, the main purpose is to show the breakdown of the system structure into its components.

Within this module we cover:

- Diagram Example and Purpose
- Blocks
- Properties
- Behaviors
- Associations
- Generalization



Internal Block Diagrams



Internal Block Diagrams (IBDs) capture the interfaces and exchanges between part and reference properties owned by a block.

Within this module we cover:

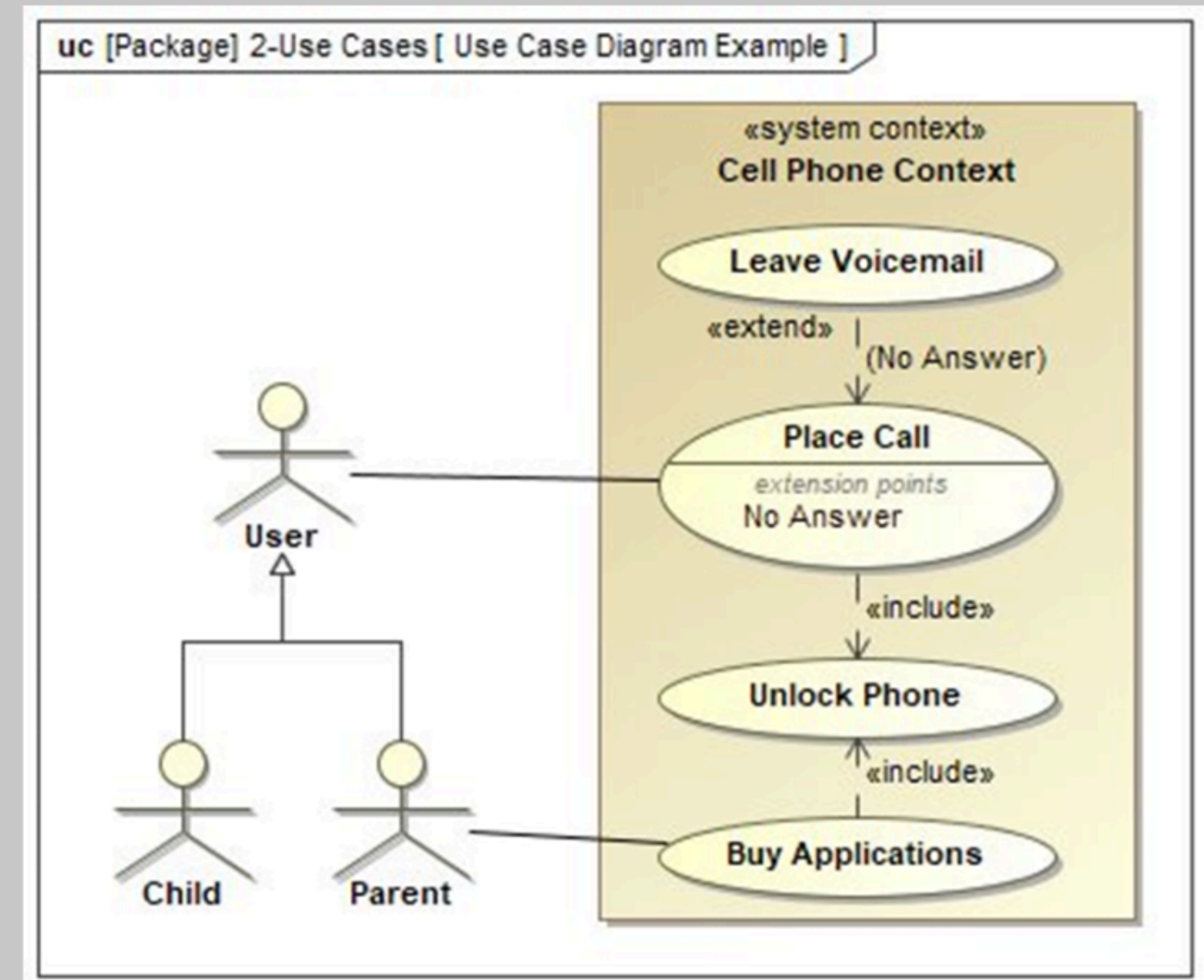
- Diagram Example and Purpose
- Interfaces
- Part/Reference/Port Properties in IBDs
- Connectors
- Item Flows

Use Cases

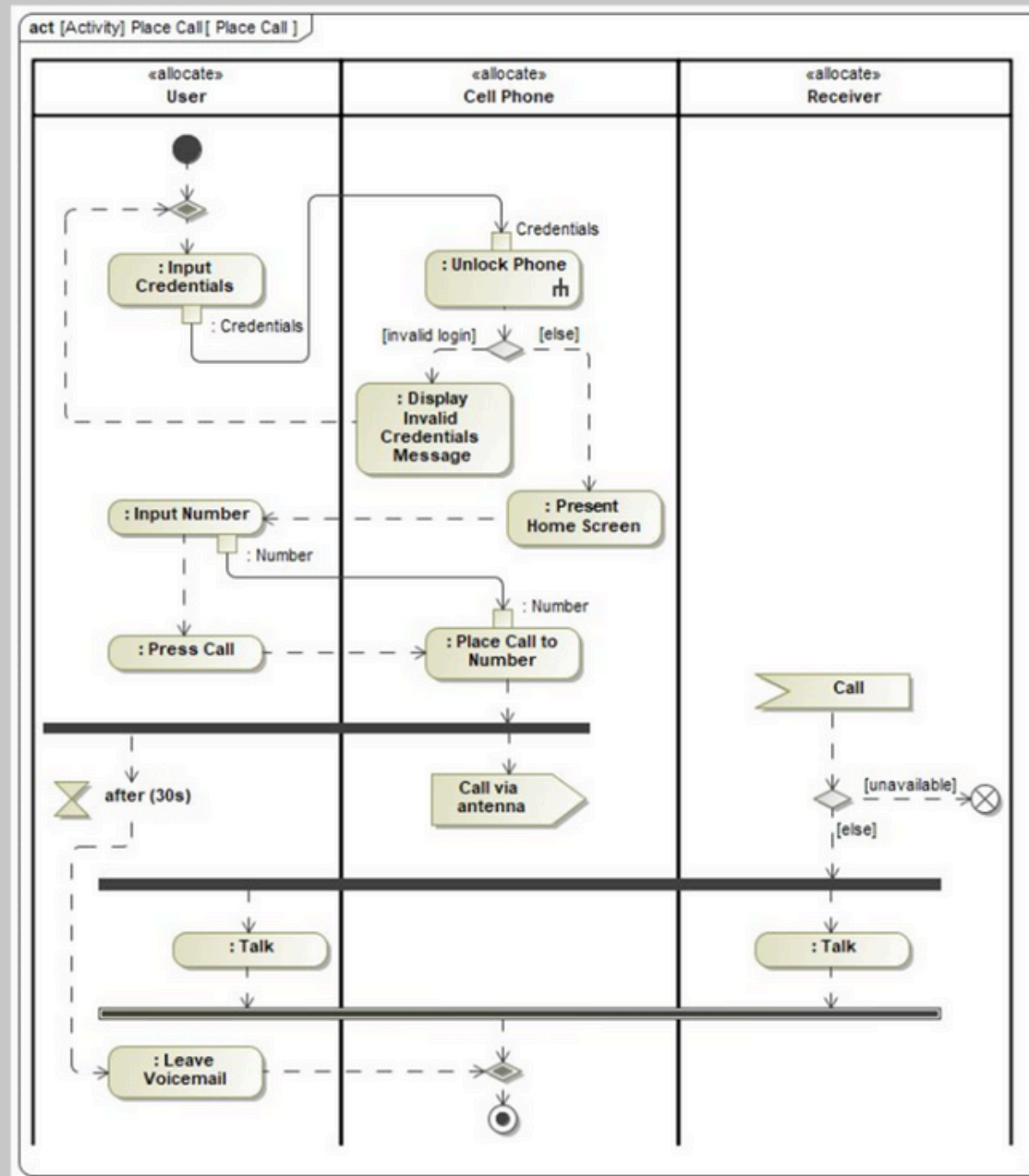
Use Cases represent the high-level, user-facing, behaviors of the system. Use Case Diagrams capture the high-level goals of the stakeholders of the system.

Within this module we cover:

- Diagram Example and Purpose
- Use Cases
- Actors
- Blocks as System Boundaries
- Associations
- Generalizations
- Includes and Extends



Activity DIAGRAMS



Activities are reusable elements that can define full system functions (placing calls), coordinate larger sets of functions (operating a phone), or represent complex steps in higher activities (ordering takeout).

Within this module we cover:

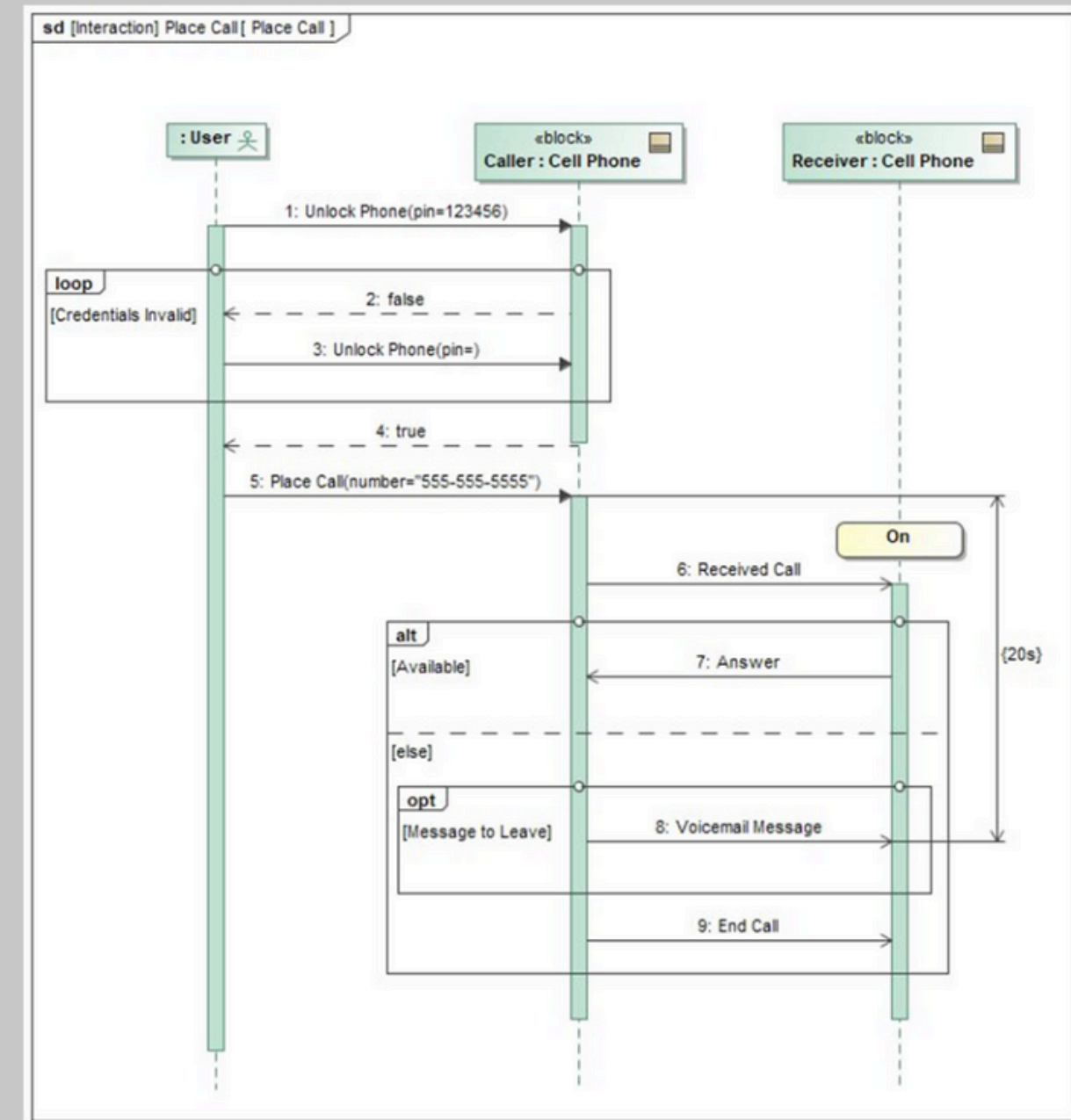
- Diagram Example and Purpose
- Use Case Connectivity
- Activity Partitions
- Actions, Flows, and Control Nodes
- Simulation Execution Rules
- Functional Decomposition
- Allocation of Behaviors to Blocks

Sequence Diagrams

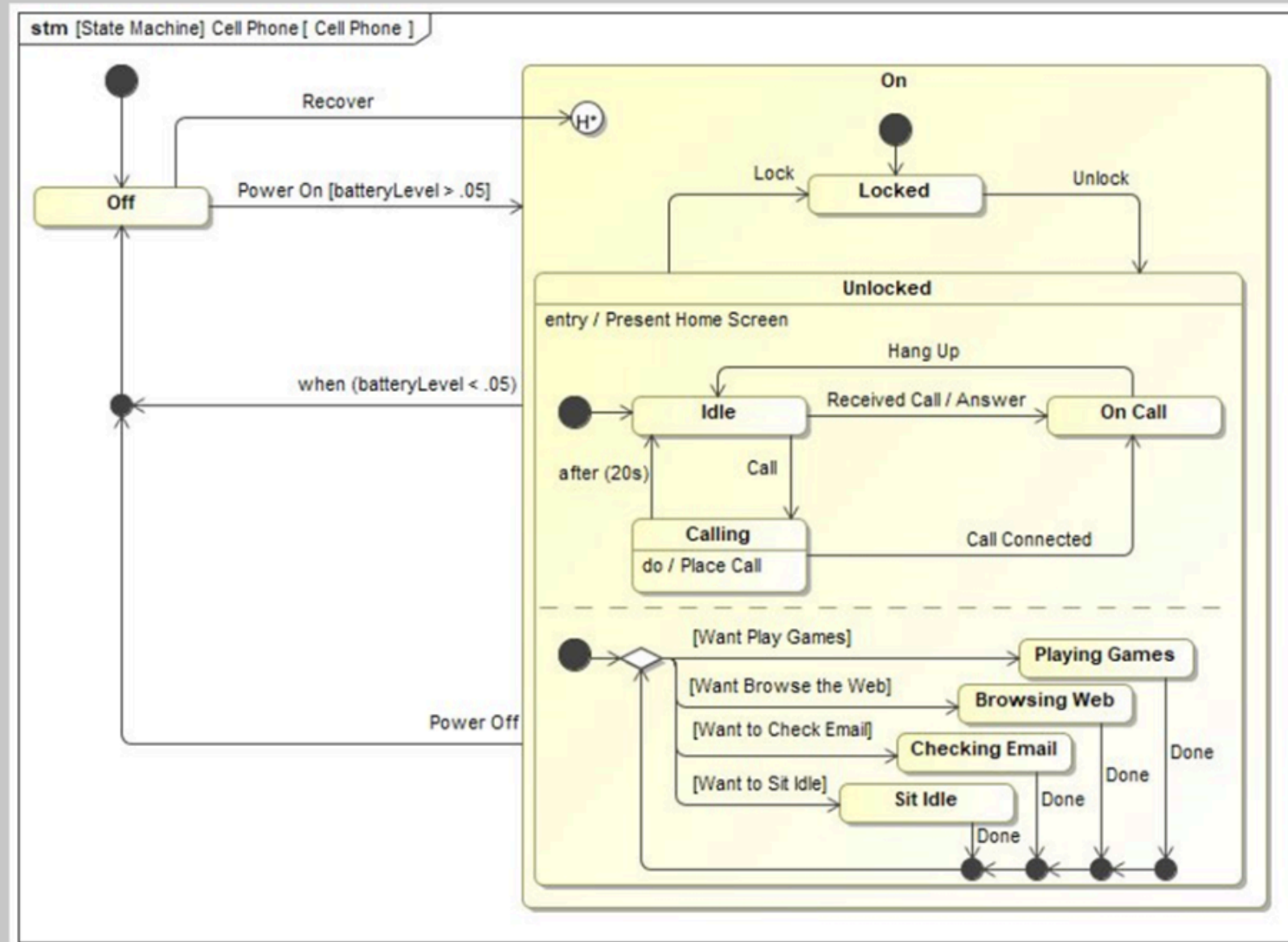
Sequence diagrams define interaction elements, which detail the sequences of message exchanges between blocks, part/reference properties, and/or actors.

Within this module we cover:

- Diagram Example and Purpose
- Sequencing
- Lifelines and Messages
- Duration/Time Constraints
- State Invariants
- Combined Fragments



State MACHINES



State Machines enable a better understanding of the significant conditions of a structure, which behaviors it can perform in these conditions, and what causes it to transition from one state to another.

Within this module we cover:

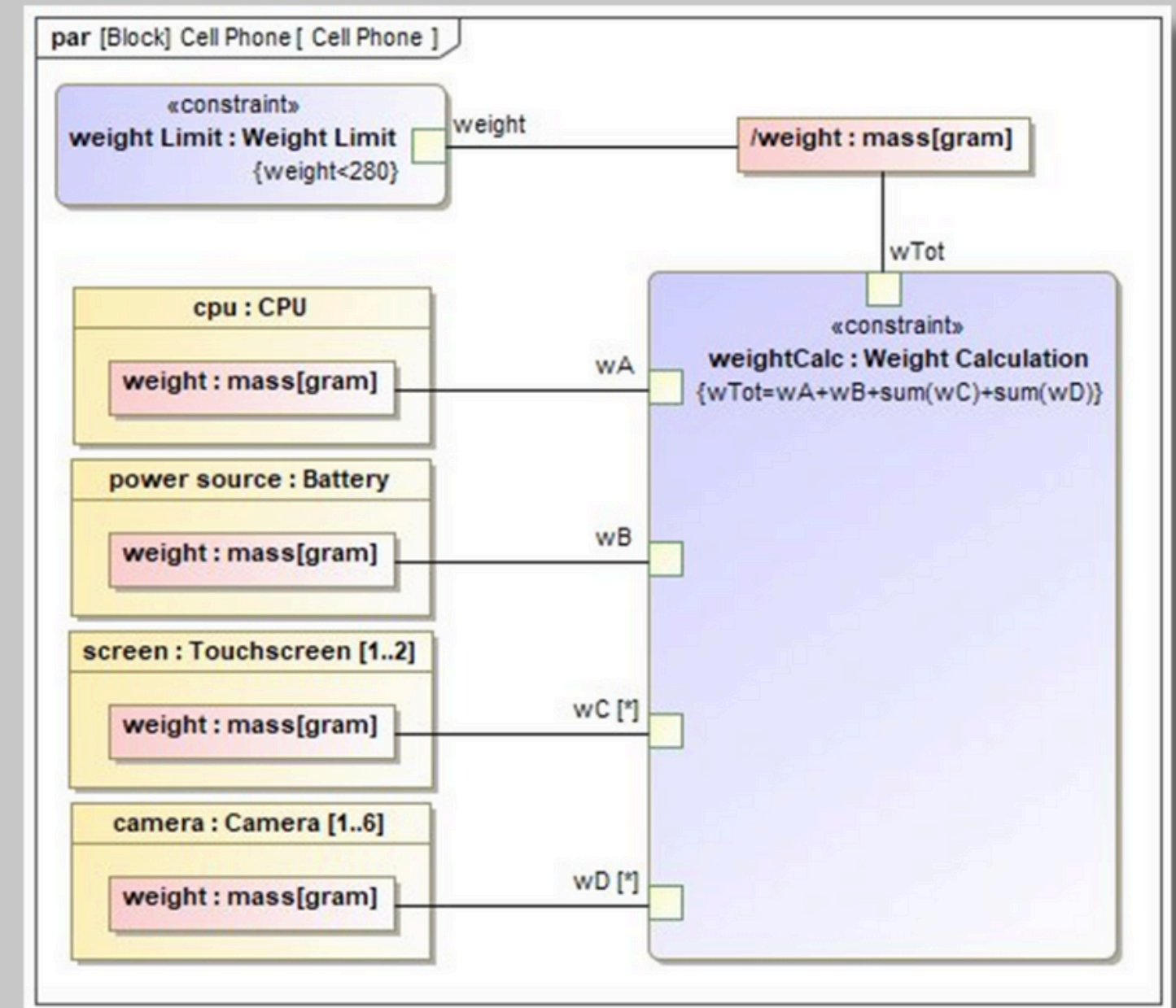
- Diagram Example and Purpose
- Navigating the State Machine
- States
- History
- Transitions
- Behaviors

Constraints & Parametrics

Constraints and Parametrics are used to define equations and connect value properties to create the mathematical model behind the system architecture

Within this module we cover:

- Diagram Example and Purpose
- Constraint Blocks
- Requirements Traceability
- Building Parametric Diagrams
- Simulation





CONTACT US

www.enola.com



training@enolatech.com



+1 877 281 7341



[linkedin.com/company/enolatech](https://www.linkedin.com/company/enolatech)

